

DotaBot rapport

étudiants : Bonzon Ludovic, Mercado Pablo & Vaz Afonso Vitor

Stack technologique

Nous avons suivi l'exemple de bot Telegram fourni par les assistants. Nous utilisons la version Javascript.

Modèle de données

Nous avons des utilisateurs créés dans MongoDB pour pouvoir avoir une référence de tous les utilisateurs Telegram ayant lié un compte Dota, car il est essentiel d'avoir ce lien lors des différentes requêtes.

Les utilisateurs sont caractérisés par :

- id: Leur ID Telegram
- username: Le nom d'utilisateur Telegram
- firstName: Prénom
- lastName: Nom de famille
- isBot: Un booléen qui indique si oui ou non l'utilisateur est un bot
- personaname: Le nom du joueur Dota associé
- accountId: L'ID du compte Steam/Dota associé

Neo4j stocke les relations ("FOLLOW") entre utilisateurs de MongoDB pour représenter qu'un utilisateur suit l'activité d'un autre utilisateur. Ces relations ne contiennent pas de champs particuliers, uniquement un ID pour les identifier.

Nous avons stockés les données des héros de Dota 2 (issus d'une requête vers l'endpoint <https://api.opendota.com/api/heroStats> de l'API OpenDota) dans un fichier javascript nommé: 'src/constants/heroes.js'.

Les héros comportent des dizaines de champs, cependant nous ne les utilisons que pour faire la traduction *hero_id* <-> *localized_name*

Requêtes

DocumentDAO (MongoDB)

- **insertUser**
Permet d'insérer les données un utilisateur Telegram lié à un compte Dota
- **getRegisteredUsers**
Permet d'obtenir les utilisateurs présents dans MongoDB

GraphDAO (Neo4j)

- **upsertUser**

Permet d'insérer ou de mettre à jour les données d'un utilisateur Telegram lié à un compte Dota

- **upsertUserFollowed**
Permet d'insérer ou de mettre à jour la relation FOLLOW entre 2 Nodes utilisateurs
- **deleteUserFollowing**
Permet de supprimer une relation FOLLOW entre 2 Nodes utilisateurs
- **getFriends**
Nous retourne toutes les Nodes au bout d'une relation FOLLOW à partir d'un user

API (OpenDota - <https://api.opendota.com/api/>)

- **getAccountId**
Permet via un *GET /search?q={query}* d'obtenir l'ID unique utilisé par Dota pour identifier les joueurs, sur la base d'un nom de joueur
- **getRecentMatchData**
Permet via un *GET /players/{account_id}/recentMatches* d'obtenir les 20 derniers matches d'un compte Dota
- **getPlayerName**
Permet via un *GET /players/{accountId}* d'obtenir le nom d'un compte Dota à partir de son ID de compte. À noter que nous récupérerons uniquement le premier compte retourné par l'appel à l'API.

Commandes

- **/help**
Une commande affichant une aide pour utiliser le bot
- **/linkaccount** <SteamID ou nom d'utilisateur Steam>
Permet de lier le compte Telegram et un compte sur Dota
- **/followplayer** <nom d'utilisateur Telegram>
Vous pouvez suivre un utilisateur du bot via son nom Telegram
- **/unfollowplayer** <nom d'utilisateur Telegram>
Vous pouvez arrêter de suivre un utilisateur du bot via son nom Telegram
- **/showfollowings**
Montre une liste des joueurs que vous suivez
- **/playeractivity** <SteamID ou nom d'utilisateur Steam>
Montre les cinq derniers matchs joués par l'utilisateur
- **/recommendhero**
Nous tirons les héros que vos joueurs suivis ont récemment joués ainsi que ceux des personnes suivies par vos joueurs suivis. Nous affichons le top 5 des héros joués par nombre d'utilisation

Note : le SteamID correspond à l'accountId utilisé par Dota.